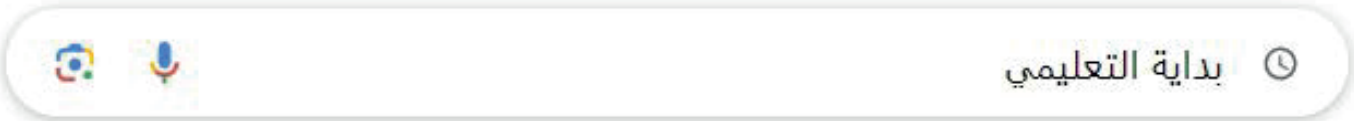
تم تحميل الملف
من موقع بداية

بداية
beadaya.com | موقع بداية التعليمي

للمزيد اكتب
في جوجل Google



موقع بداية التعليمي كل ما يحتاجه **الطالب والمعلم**
من ملفات تعليمية، حلول الكتب، توزيع المنهج،
بوربوينت، اختبارات، ملخصات، اختبارات إلكترونية،
أوراق عمل، والكثير...

حمل التطبيق

GET IN ON Google Play

Download on the App Store

رابط الدرس الرقمي

www.ien.edu.sa

الدرس الأول:

التحكم في الروبوت (Robot Control)

المتغيرات

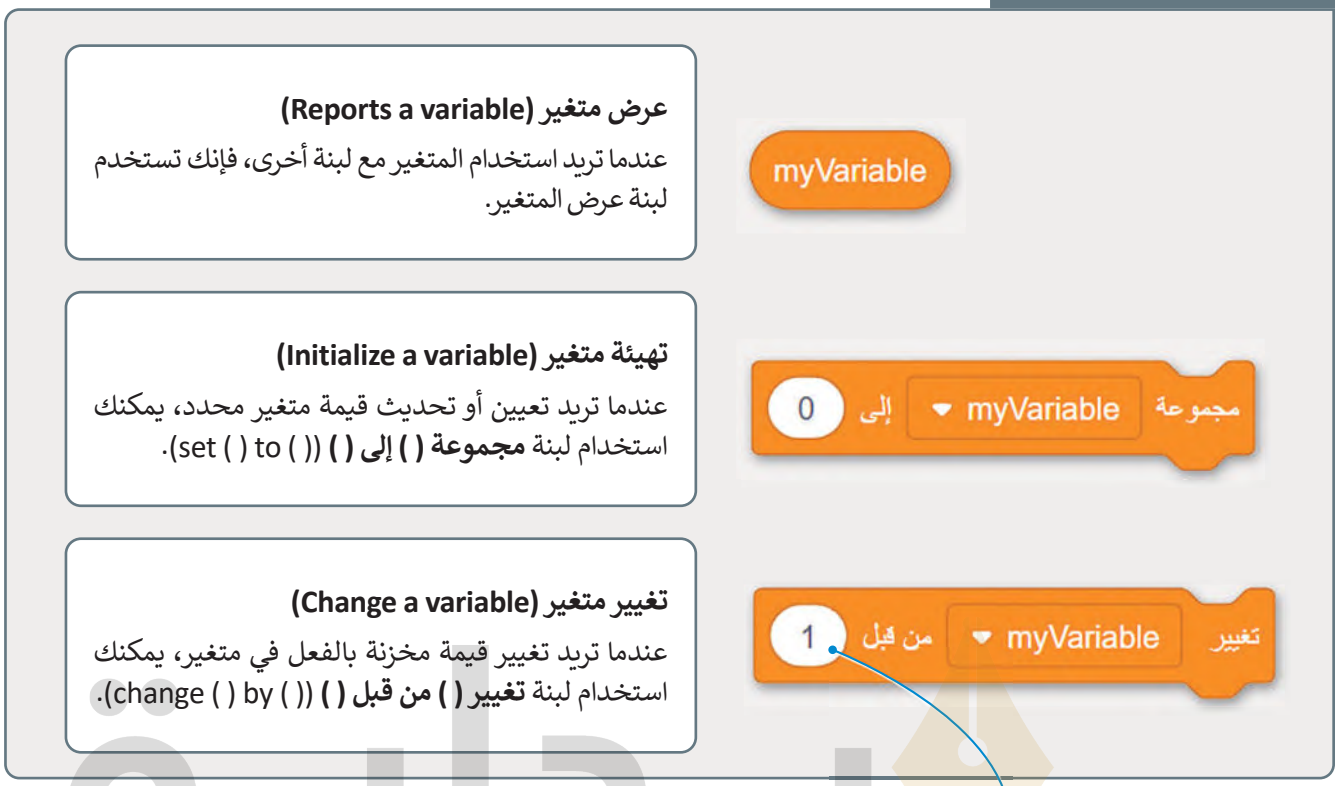
المتغير يشبه الاسم المستعار لشيء يجب أن يتذكره جهاز الحاسب. تعمل المتغيرات مثل الحاويات في البرامج للحفاظ على البيانات التي يمكن أن تكون أرقامًا وأحرفًا.

لتخزين أنواع مختلفة من البيانات، هناك فئتان رئيستان من المتغيرات وهما: المتغيرات الرقمية والمتغيرات النصية، ويطلق على المتغيرات النصية أيضًا اسم **السلاسل النصية** (Strings).

يمكنك العثور على جميع اللبنات الخاصة بالمتغيرات في **فئة المتغيرات** (Variables).

beadaya.com موقع بداية التعليمي

وصف لبنات المتغير



beadaya.com | موقع بداية التعليمي

اسم المتغير

عندما تنشئ متغيرًا فإنك تحدد اسمه.

> يجب أن يكون اسم كل متغير فريدًا ولم يستخدم سابقًا في نفس البرنامج.

> يمكن أن يتكون اسم المتغير من مجموعة أحرف كبيرة وصغيرة، ويمكنك استخدام أكثر من كلمة مع وجود **شرطة سفلية** (_) بينهما.

> بعض الكلمات لا يمكن استخدامها كاسم متغير؛ لأنها كلمات خاصة تستخدمها بالفعل بيئة البرمجة (على سبيل المثال: تكرار، محرك الأقراص، الدوران، بينما، إذا، آخر، إلخ). وتسمى بالكلمات الرئيسة المحجوزة.

> يجب ألا يحتوي اسم المتغير على أحرف خاصة (على سبيل المثال: !، "، إلخ)، وأيضًا لا يبدأ برقم ولا يحتوي على مسافات.

> يفضل أن يمثل اسم المتغير محتواه؛ حتى تفهم ما يمثله المتغير عندما تراه في الكود.

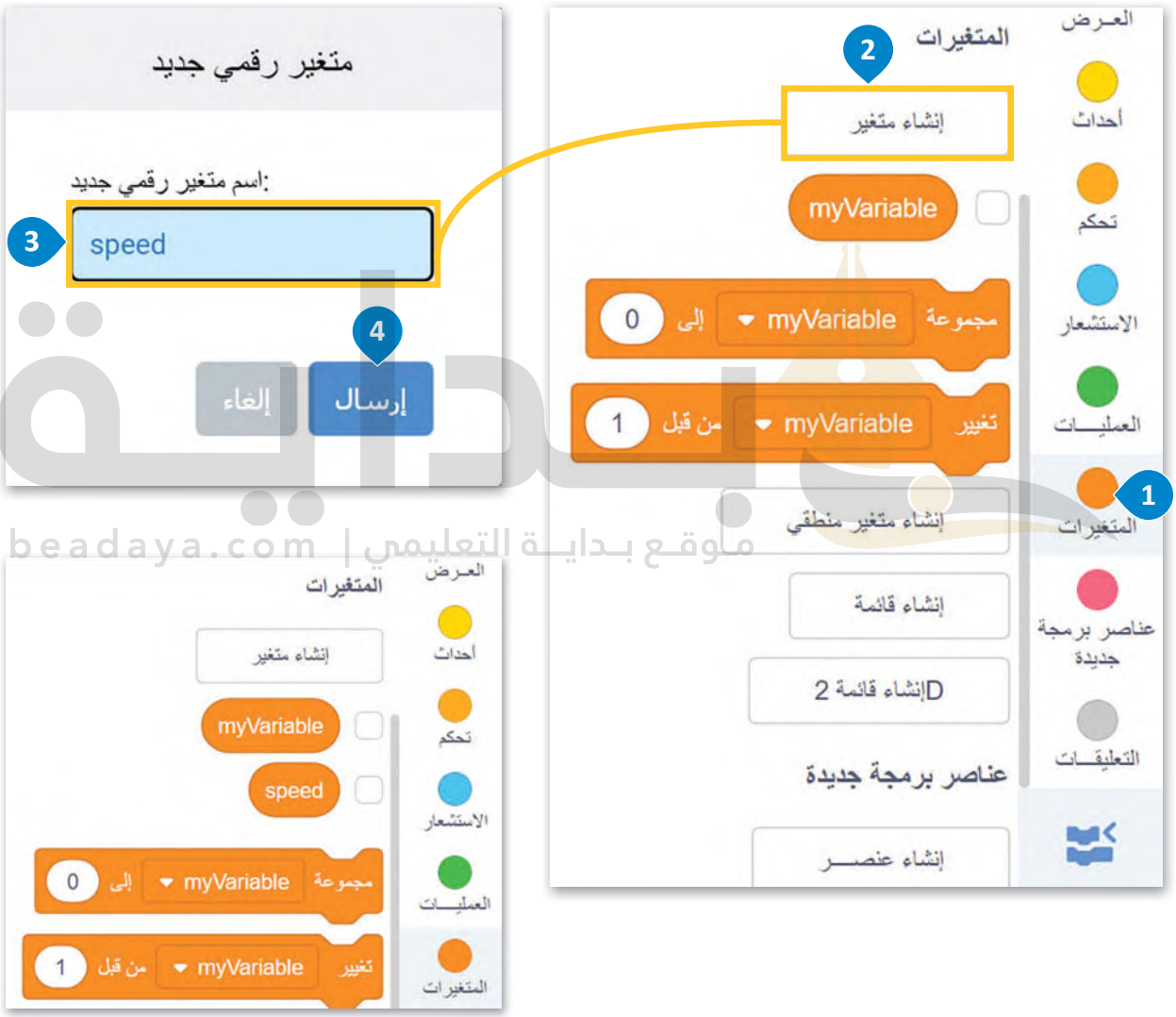
إنشاء متغير رقمي

عليك إنشاء متغير قبل استخدامه في بيئة فيكس كود في آر، أنشئ متغيرًا رقميًّا جديدًا.

لإنشاء متغير رقمي:

> من فئة **المتغيرات** (Variables)، 1 اضغط على **إنشاء متغير** (Make a Variable). 2

> في نافذة **متغير رقمي جديد** (New Numeric Variable)، اكتب اسمًا للمتغير، على سبيل المثال "speed"، 3 ثم اضغط على **إرسال** (Submit). 4



beadaya.com | موقع بداية التعليمي

إعادة تسمية متغير رقمي

يمكنك إعادة تسمية كل متغير في بيئة فيكس كود في آر، أعد تسمية المتغير الافتراضي "myVariable".

لإعادة تسمية myVariable:

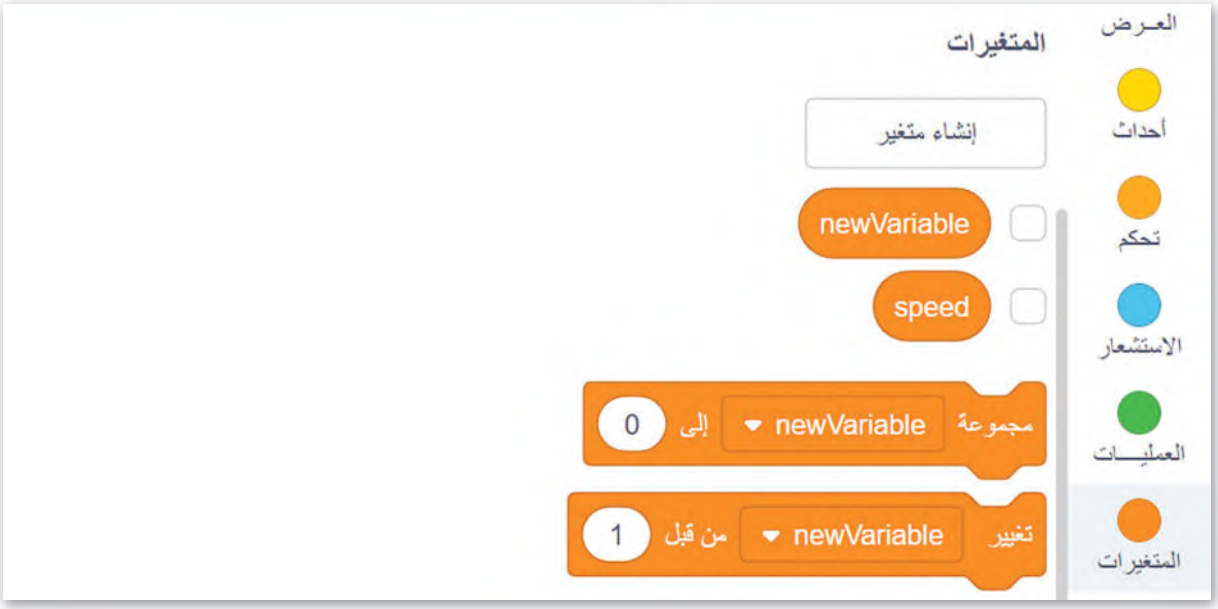
> من فئة **المتغيرات** (Variables) 1 اضغط بزر الفأرة الأيمن على لبنة **myVariable**. 2

> من القائمة المنسدلة، اختر **إعادة تسمية المتغير** (Rename variable). 3

> في النافذة **إعادة تسمية المتغير** (Rename variable)، اضغط على لبنة **myVariable** 4 واكتب الاسم الجديد للمتغير، على سبيل المثال **"newVariable"**. 5 واضغط على **إرسال** (Submit). 6

بداية
beadaya.com | موقع بداية التعليمي

تم تغيير اسم المتغير إلى (newVariable).



حذف متغير (Delete variable)

يمكنك حذف المتغير الافتراضي في بيئة فيكس كود في آر، احذف المتغير "newVariable".

لحذف متغير:

> من فئة **المتغيرات** (Variables)، 1 اضغط بزر الفأرة الأيمن على لبنة **newVariable**. 2

> من القائمة المنسدلة، اختر **حذف المتغير "newVariable"** (Delete the "newVariable" variable). 3

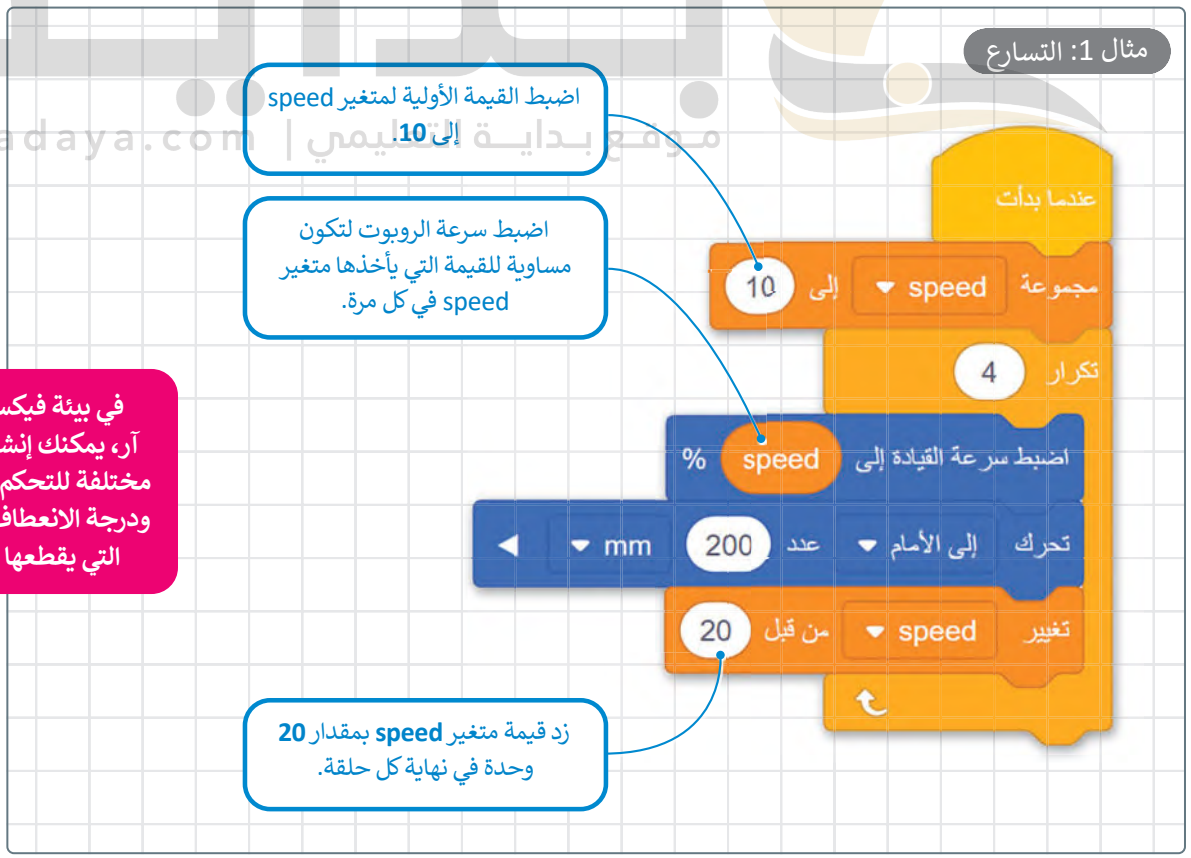
beadaya.com | موقع بداية التعليمي

الآن، في فئة المتغيرات هناك متغير speed.

طريقة استخدام المتغيرات للتحكم في حركات روبوت الواقع الافتراضي

باستخدام متغير speed، ستشاهد كيف يمكنك الاستفادة من استخدامه في بيئة فيكس كود في آر.

باستخدام **شبكة خريطة** (Grid Map)، يمكنك اختبار روبوت الواقع الافتراضي في المثال التالي، حيث يبدأ الروبوت في التحرك للأمام بسرعة 10 %. باستخدام متغير speed ، يمكنك جعل الروبوت يتسارع بنسبة 20 % كل 200 **مليمتر** (mm).



beadaya.com | موقع بداية التعليمي

مراقبة قيم المتغير

يمكنك فتح وحدة **تحكم المراقبة** (Monitor Console) لعرض التغييرات التي تحدث للمتغير speed عند تشغيل البرنامج. ألق نظرة على كيفية تغير قيم المتغير speed أثناء مرحلة التسارع.

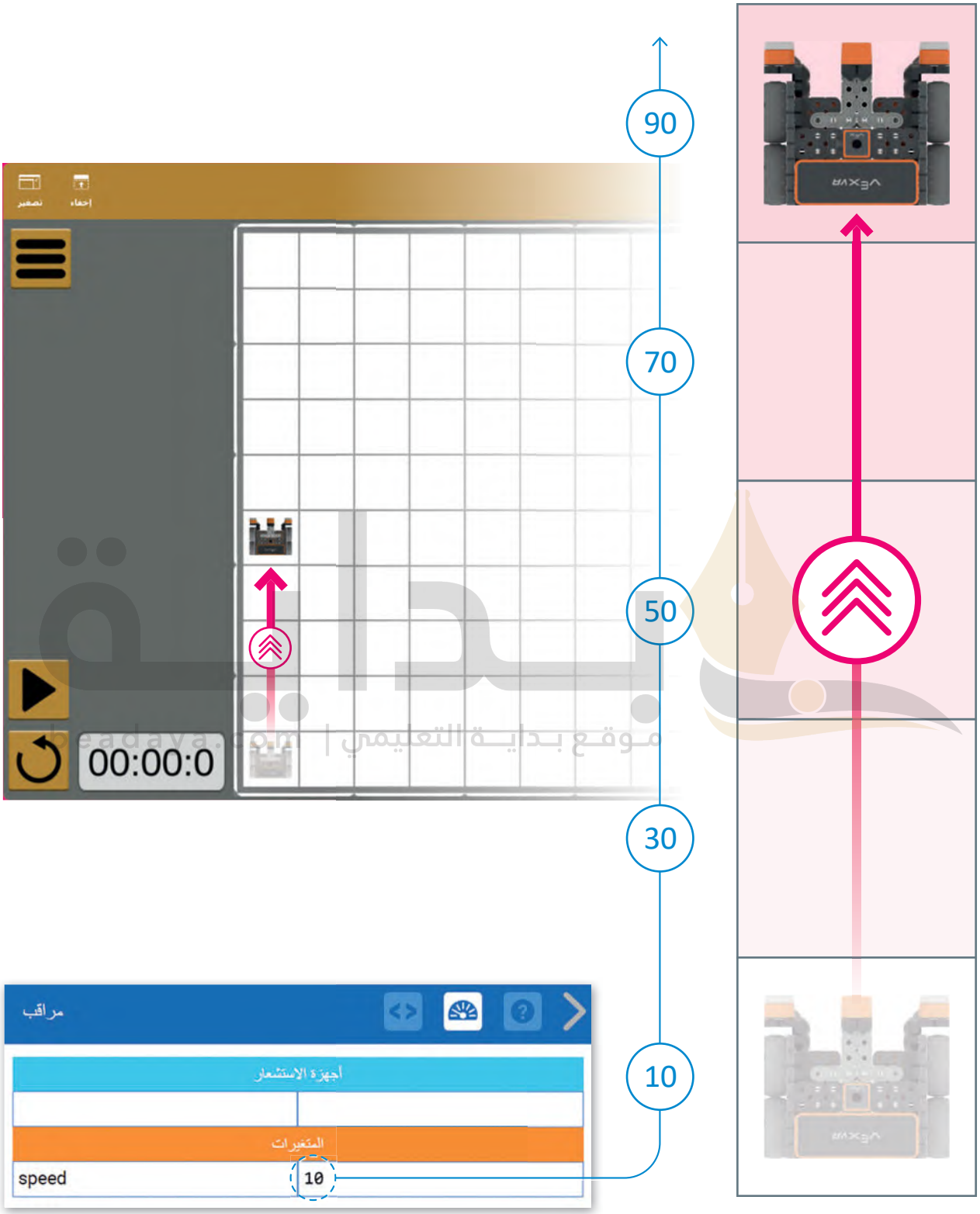
لتنفيذ البرنامج:

> من فئة **المتغيرات** (Variables)، 1 حدد خانة الاختيار متغير speed. 2

> حدد أيقونة **تحكم المراقبة** (Monitor Console). 3

> اضغط على **بداية** (Start). 4

beadaya.com | موقع بداية التعليمي



بداية
موقع بداية التعليمي | beaddaya.com

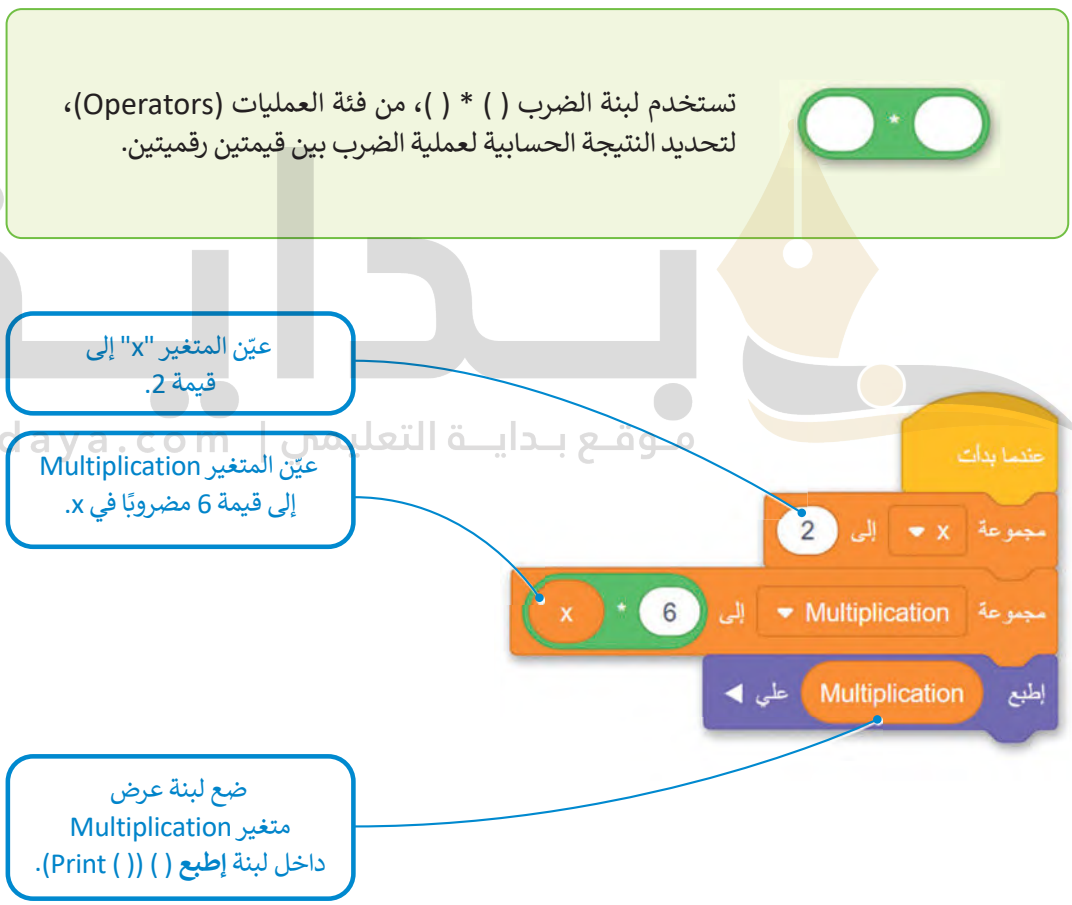
العمليات الحسابية

في البرمجة تستخدم المعاملات الرياضية لإجراء الحسابات. يمكنك استخدام فيكس كود في آر لإجراء أي عملية حسابية مثل: الجمع، والطرح، والضرب، والقسمة، وغيرها.

كما تعلمت سابقًا، المُعامِل هو رمز يمثل إجراءً محددًا، على سبيل المثال: علامة **الجمع** (+) هي مُعامِل يمثل الجمع. وتسمى المُعامِلات التي تستخدمها لإجراء العمليات الحسابية بالمُعامِلات الرياضية، ويمكنك العثور على المُعامِلات الرياضية في فئة **العمليات** (Operators).

مثال 2: العمليات الحسابية

في المثال التالي، ستنفّذ عملية حسابية بسيطة في بيئة فيكس كود في آر. ستستخدم المتغير "x" الذي ستعيّنه إلى قيمة 2. ستستخدم أيضًا متغير "Multiplication" الذي ستعيّنه إلى قيمة متغير "x" مضروبًا في 6، باستخدام بيئة **عملية الضرب** (multiplication operator).



بداية
beadaya.com | موقع بداية التعليمي

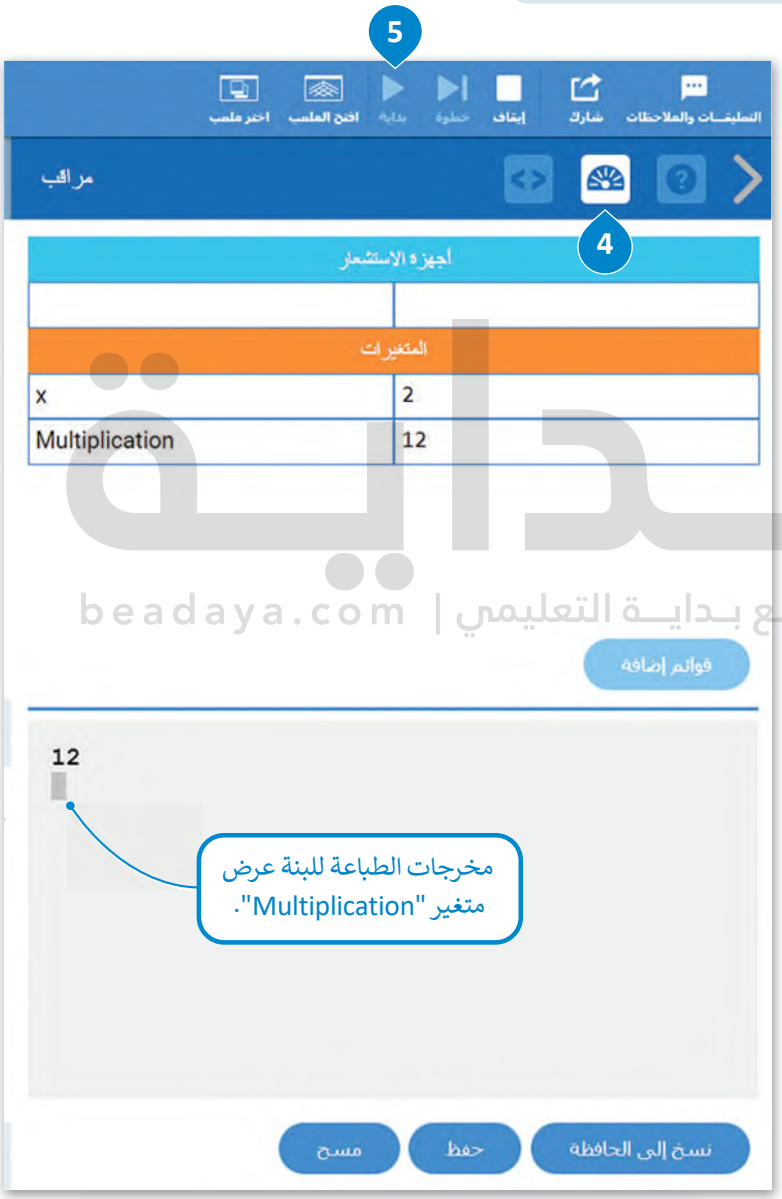
عند تنفيذ البرنامج تتم مراقبة قيم المتغيرات "x" و "Multiplication" عن طريق **وحدة تحكم المراقبة** (Monitor Console) ويتم طباعة قيمة متغير Multiplication إلى **وحدة تحكم العرض** (Print Console).

لتنفيذ البرنامج:

> من فئة **المتغيرات** (Variables)، 1 حدد **خانة الاختيار** (checkbox) للمتغير x، 2 **وخانة الاختيار** (checkbox) للمتغير Multiplication. 3

> حدد رمز **وحدة تحكم المراقبة** (Monitor Console). 4

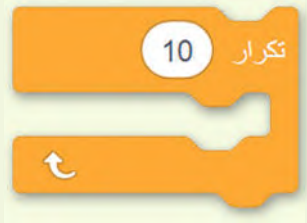
> اضغط على زر **بداية** (Start). 5



بداية
موقع بداية التعليمي | beadaya.com

قد ترغب أحيانًا في تنفيذ نفس التعليمات البرمجية عدة مرات، حتى تتمكن من استخدام **التكرارات** (Loops)، والتي تسمح لك بتكرار نفس الأوامر عدة مرات. يوفر فيكس كود في آر أربعة أنواع من التكرارات وهي: **تكرار** (repeat)، و**تكرار حتى** (repeat until)، و**إلى الأبد** (forever) و**في حين** (while).

تكرار ( ) مرات (Repeat ( ) times)



لبنة **التكرار ( )** ، تستخدم عند تنفيذ اللبنات البرمجية الموجودة لعدد محدد سابقًا من المرات.

مثال 3: العمليات الحسابية في تكرارات

في المثال التالي، ستنفّذ عملية حسابية 10 مرات باستخدام حلقة **تكرار ( ) مرات** (Repeat ( ) times). ستعيّن المتغير "x" في البداية يساوي 0 وستبرمجه ليتم زيادته بمقدار 1 في كل مرة يتم فيها تنفيذ التكرار. ستعيّن متغير "Multiplication" الذي يساوي المتغير "x" مضروبًا في 6، باستخدام بيئة عمليات الضرب. في كل مرة يتم فيها تنفيذ التكرار، يتم تحديد قيمة متغير "Multiplication" بواسطة القيمة الحالية للمتغير "x" مضروبة في 6.

بداية
beadaya.com | موقع البداية التعليمي

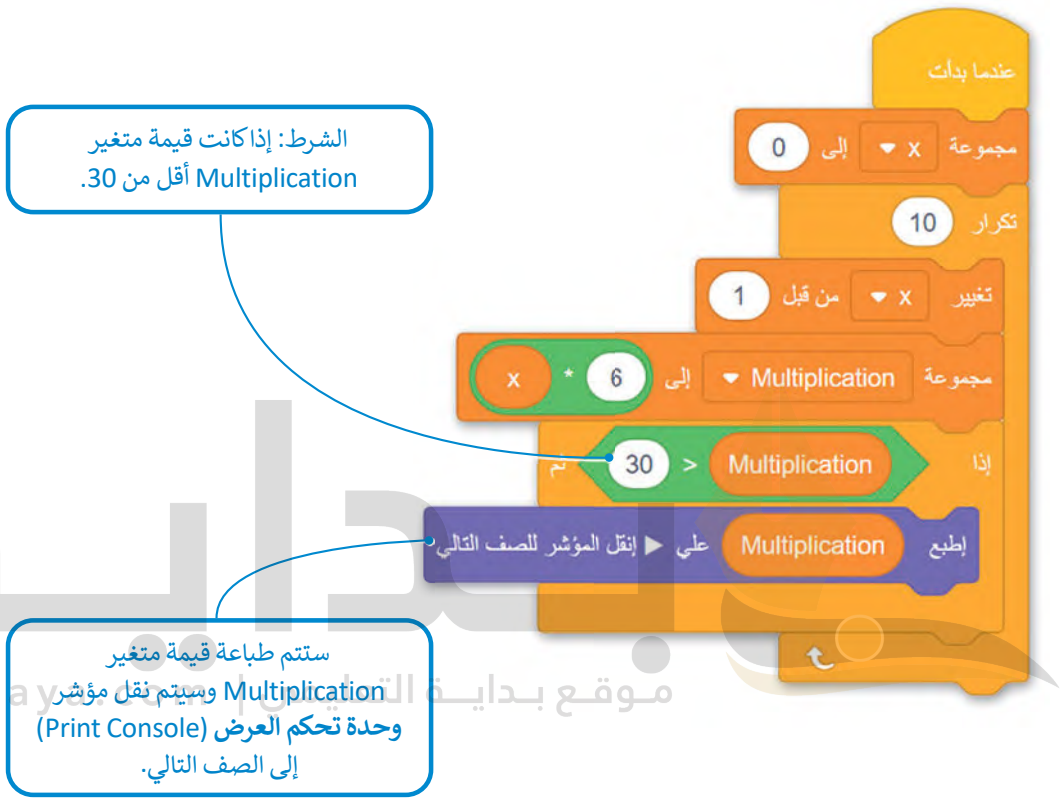
أثناء تنفيذ البرنامج، تكون النتيجة هي مخرجات في **وحدة تحكم العرض** (Print Console).

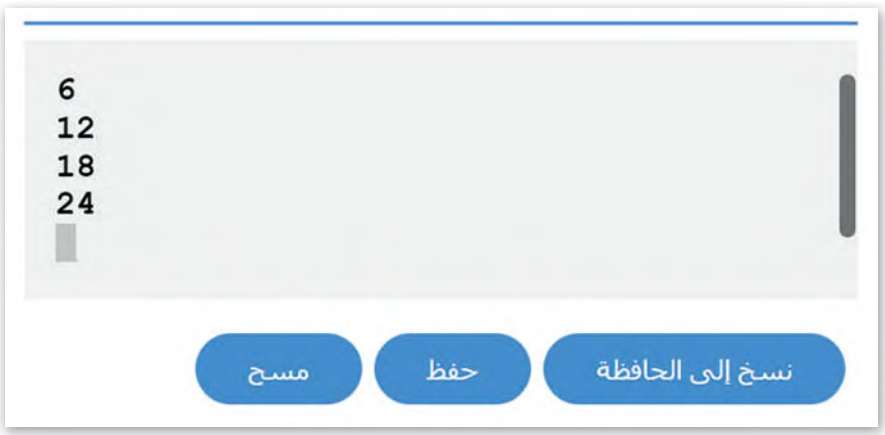
لا تنس استخدام زر مسح (CLEAR) وإلا فسيتم الاحتفاظ بالرسائل في وحدة تحكم العرض (Print Console) بعد تنفيذ البرنامج.

بداية
beadaya.com | موقع بداية التعليمي

مثال 4: العمليات الحسابية واستخدام الشرطية في الحلقات

في المثال السابق، ستضيف، داخل الحلقة، لبنة **إذا ( )** (( ) if) لفئة **التحكم** (Control) للتحقق مما إذا كان الشرط صحيحًا، عند كل تكرار. إذا كان الشرط صحيحًا، يتم تنفيذ أمر اللبنة داخل لبنة **إذا ( ).** على وجه التحديد، يتحقق هذا الجزء من التعليمات البرمجية عند كل تكرار، إذا كانت قيمة متغير "Multiplication" أقل من 30. إذا كان هذا صحيحًا، فإن قيمة متغير "Multiplication" يتم إخراجها في **وحدة تحكم العرض** (Print Console). لبرمجة حالة لبنة **إذا ( )**، ستستخدم لبنة **( ) أقل من ( )** من فئة **العمليات** (Operators).





بداية
موقع بداية التعليمي
beaday

لبنة تكرار حتى (Repeat Until)

في بعض الأحيان تريد تنفيذ برنامج حتى يكون شرط معين صحيحًا. للقيام بذلك، يمكنك استخدام لبنة **تكرار حتى** (Repeat Until). تتيح لك الحلقة الشرطية تشغيل البرنامج عدة مرات بينما يظل الشرط خطأً.

لبنة **تكرار حتى ( )** هي لبنة يتم فيها التكرار طالما أن الشرط خطأ، وعندما يصبح الشرط صحيحًا، سيتوقف تنفيذ التكرار.

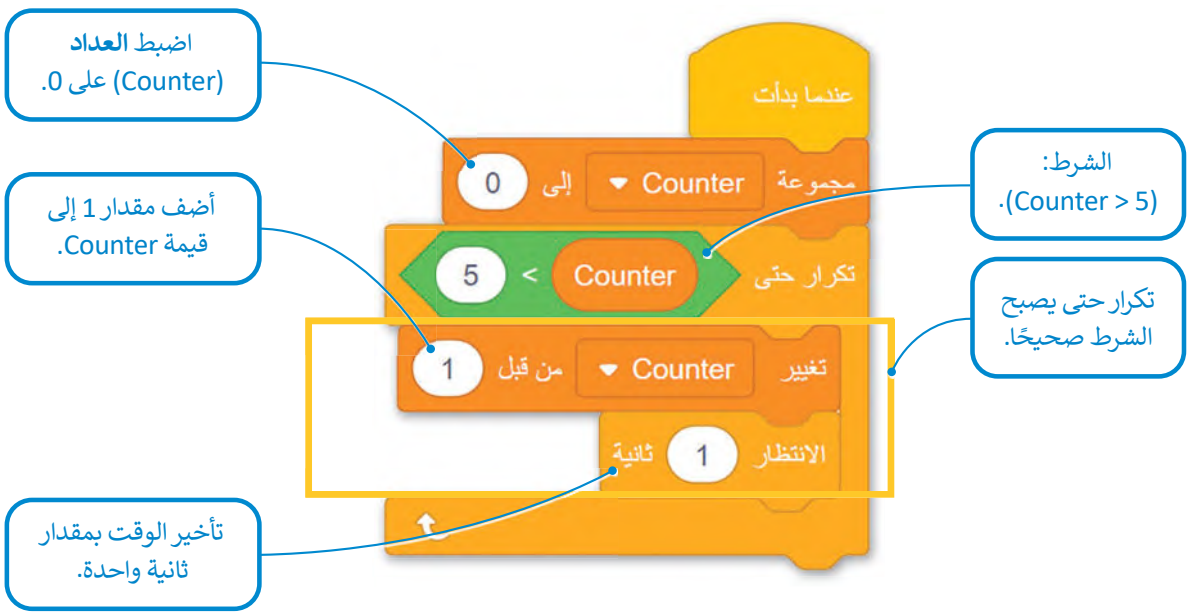
في العديد من الحالات، تريد أن يعتمد هذا الشرط على عدد المرات التي يتم فيها تنفيذ التكرار. لحساب عدد تكرارات جزء من التعليمات البرمجية، يمكنك استخدام متغير رقمي يسمى **العداد** (Counter). يمكنك تعريف القيمة الأولية للعداد، كما يمكنك تحديد القيمة التي تتغير من تكرار إلى آخر، كما يمكنك برمجة الشرط الذي يتحكم في التكرار باستخدام **العداد** (Counter).

في هذه الحالة، عندما يكون للعداد قيمة معينة، يصبح شرط حلقة **تكرار حتى ( )** صحيحًا عندها يتوقف التكرار.

مثال 5: العدّ

في المثال التالي، ستبرمج **Counter** ويتم تعيينه على 0 في بداية البرنامج، وليتم زيادته بمقدار 1 في كل مرة يتم فيها تنفيذ تكرار داخل تكرار، ستضيف لبنة **الانتظار**(wait) بقيمة زمنية مدتها 1 ثانية. وأخيرًا، سوف تستخدم لبنة **أكبر من ( )** من فئة العمليات لبرمجة حالة حلقة **تكرار حتى ( )**. عندما يصبح العداد أكبر من 5، تتوقف التكرارات.

beadaya.com | موقع بداية التعليمي



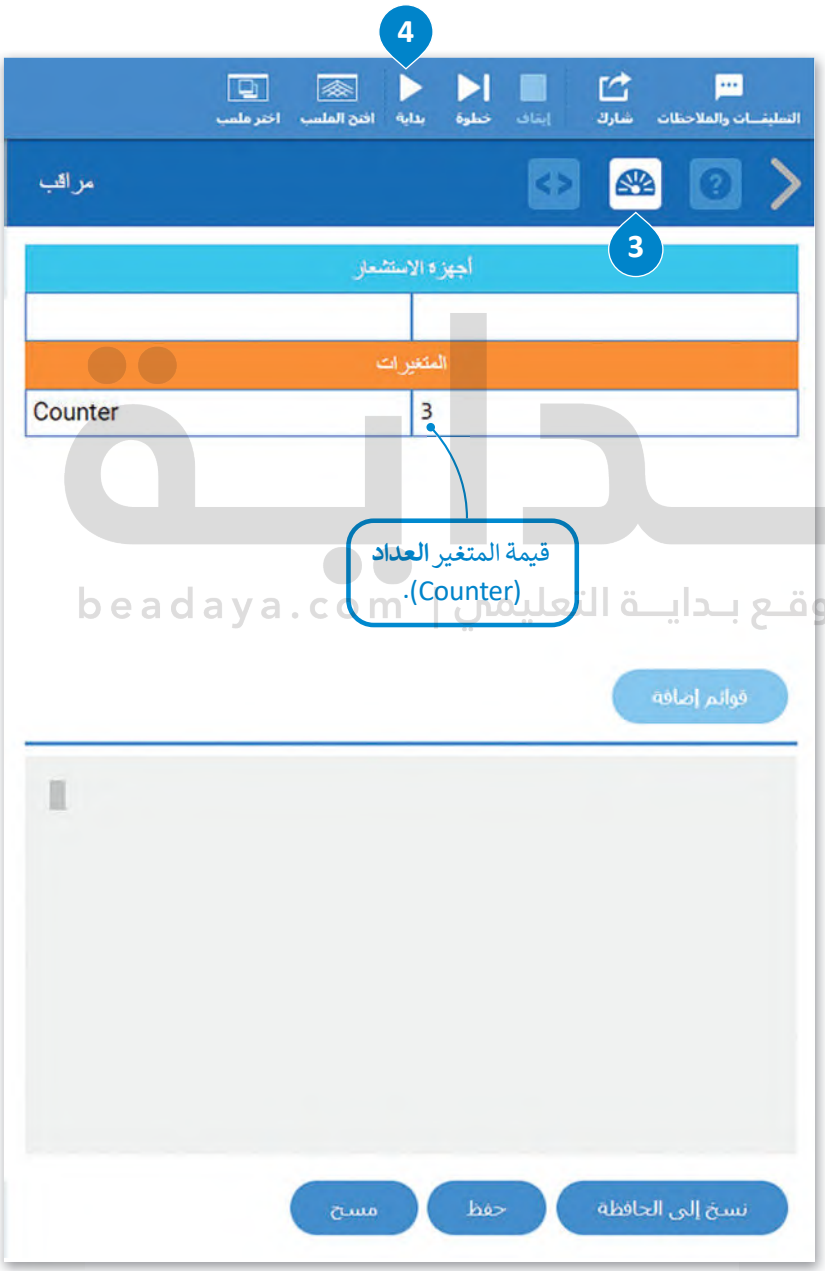
أثناء تنفيذ البرنامج يمكنك مراقبة متغير **العداد** (Counter) ليتم زيادته على التوالي من 1 إلى 5، في وحدة **تحكم المراقبة** (Monitor Console).

لتنفيذ البرنامج:

> من فئة **المتغيرات** (Variables)، 1 حدد **خانة الاختيار** (checkbox) لمتغير **العداد** (Counter). 2

> اختر أيقونة **وحدة تحكم المراقبة** (Monitor Console). 3

> اضغط على **بداية** (Start). 4

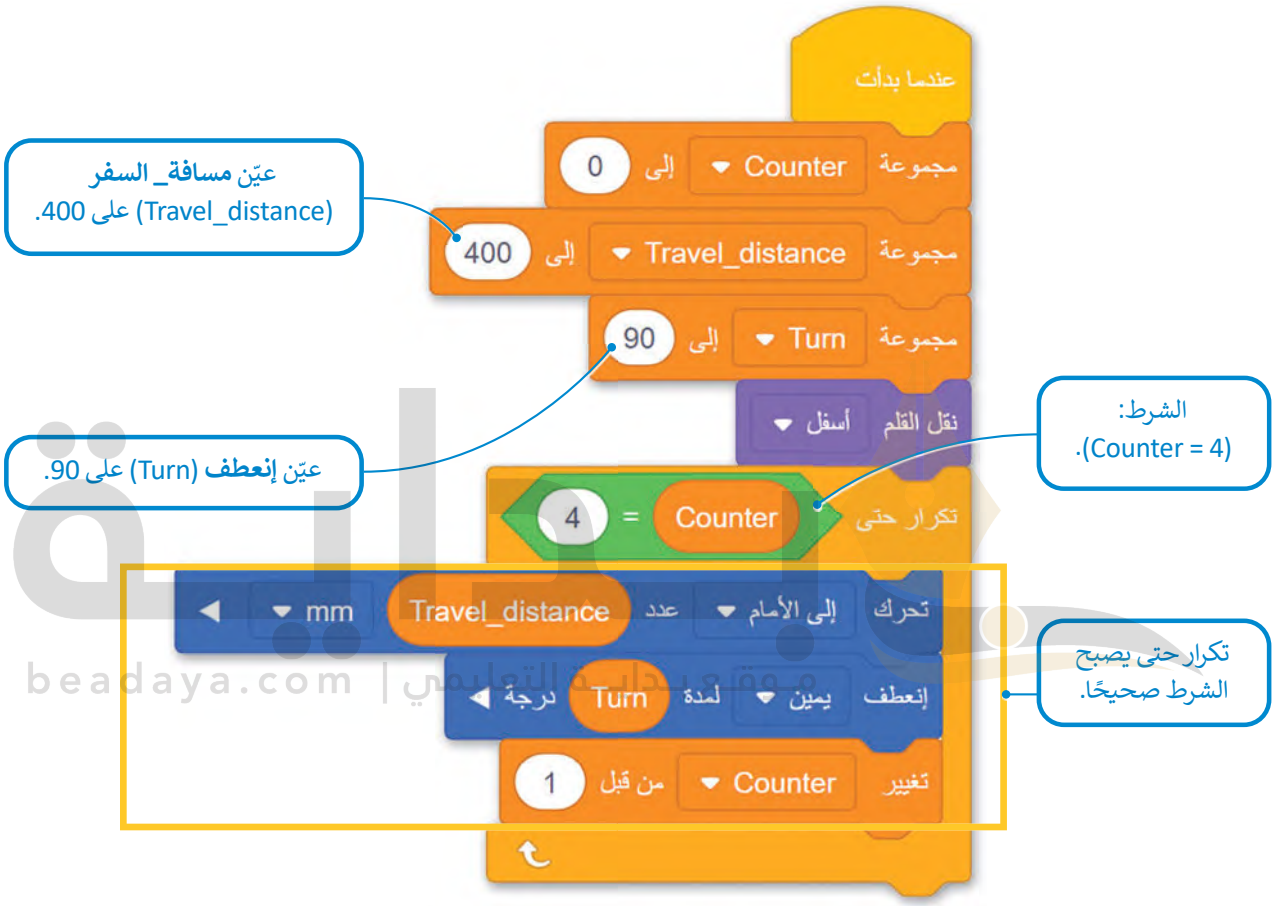


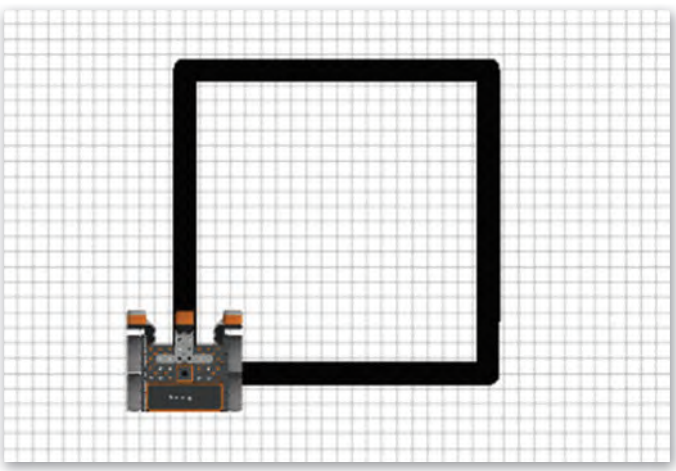
beadaya.com | موقع بداية التعليمي

مثال 6: رسم مربع

في المثال التالي، ستبرمج روبوت الواقع الافتراضي لرسم مربع في ملعب **الفن قماش** (Art Canvas)، باستخدام حلقة **تكرار حتى** (repeat until)، ومتغير إضافة 1 إلى القيمة المقابلة ستبرمج الحلقة للتكرار 4 مرات، سيتم تعيين العداد في البداية إلى 0 وسيتم زيادته بمقدار 1 في كل تكرار، حتى يأخذ القيمة 4 ، وهذا هو الشرط الذي سيوقف فيه التكرارات.

في حين أن قيمة العداد هي 0 و 1 و 2 و 3 ، فإن روبوت الواقع الافتراضي يتحرك إلى الأمام لمسافة تساوي قيمة المتغير **مسافة_ السفر** (Travel_ distance) ويجعل الانعطافات لليمين مساوية لقيمة المتغير **إنعطف** (Turn). يتم تعريف قيم هذين المتغيرين في بداية البرنامج.





beadaya.com | بداية للتعليمي

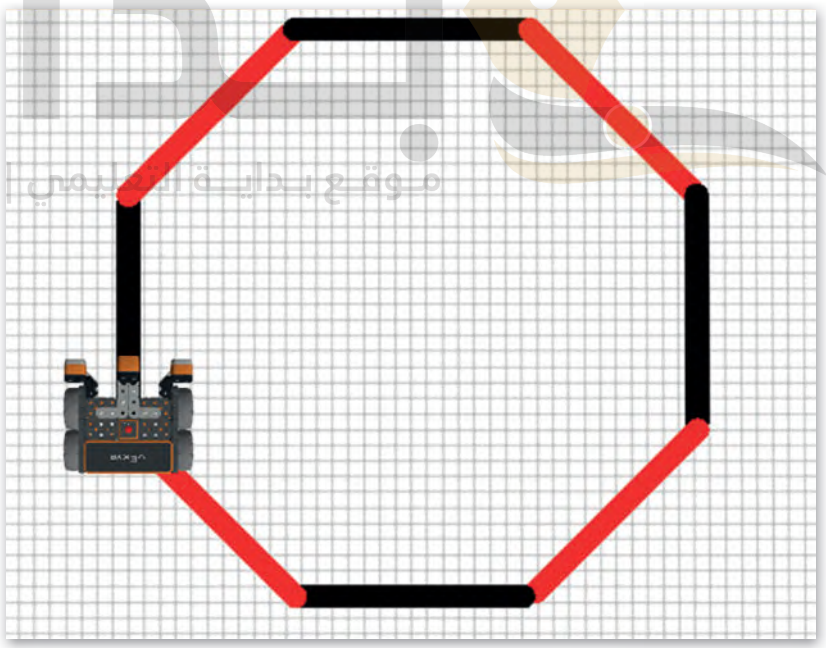
الأعداد الزوجية والفردية

في بعض الأحيان تريد التمييز بين نتيجة البرنامج اعتمادًا على عدد حلقة التكرارات. إذا كان رقم التكرار عددًا فرديًّا، فأنت تبرمج نتيجة معينة. وإذا كان رقم التكرار عددًا زوجيًّا، فأنت تبرمج نتيجة مختلفة. للقيام بذلك، يجب عليك استخدام متغير **العداد** (Counter) عند الشرط الذي ينهي الحلقة **تكرار حتى** ( ). عندما يأخذ متغير Counter قيمة معينة يتم إنهاء البرنامج. حتى ذلك الحين، إذا كان متغير Counter عددًا فرديًا، فإن البرنامج لديه نتيجة معينة وإذا كان متغير Counter رقمًا زوجيًا، فإن البرنامج لديه نتيجة مختلفة.

مثال 7: رسم شكل ثماني

في المثال التالي، برمج روبوت الواقع الافتراضي لرسم شكل ثماني في ملعب **الفن قماش** (Art Canvas) وغيّر لون أداة القلم اعتمادًا على عدد متغير Counter. إذا كان المتغير Counter عددًا زوجيًا، فإنه يغير لون أداة القلم إلى الأسود، وإذا كان المتغير Counter فرديًا فإنه يغيرها إلى الأحمر. لإنشاء الشرط الذي سيحدد ما إذا كانت قيمة العداد هي رقم زوجي أو فردي، ستستخدم لبنة **ما تبقى من ( ) / ( )** (remainder of ( ) / ( )) للعداد مقسومة على 2.

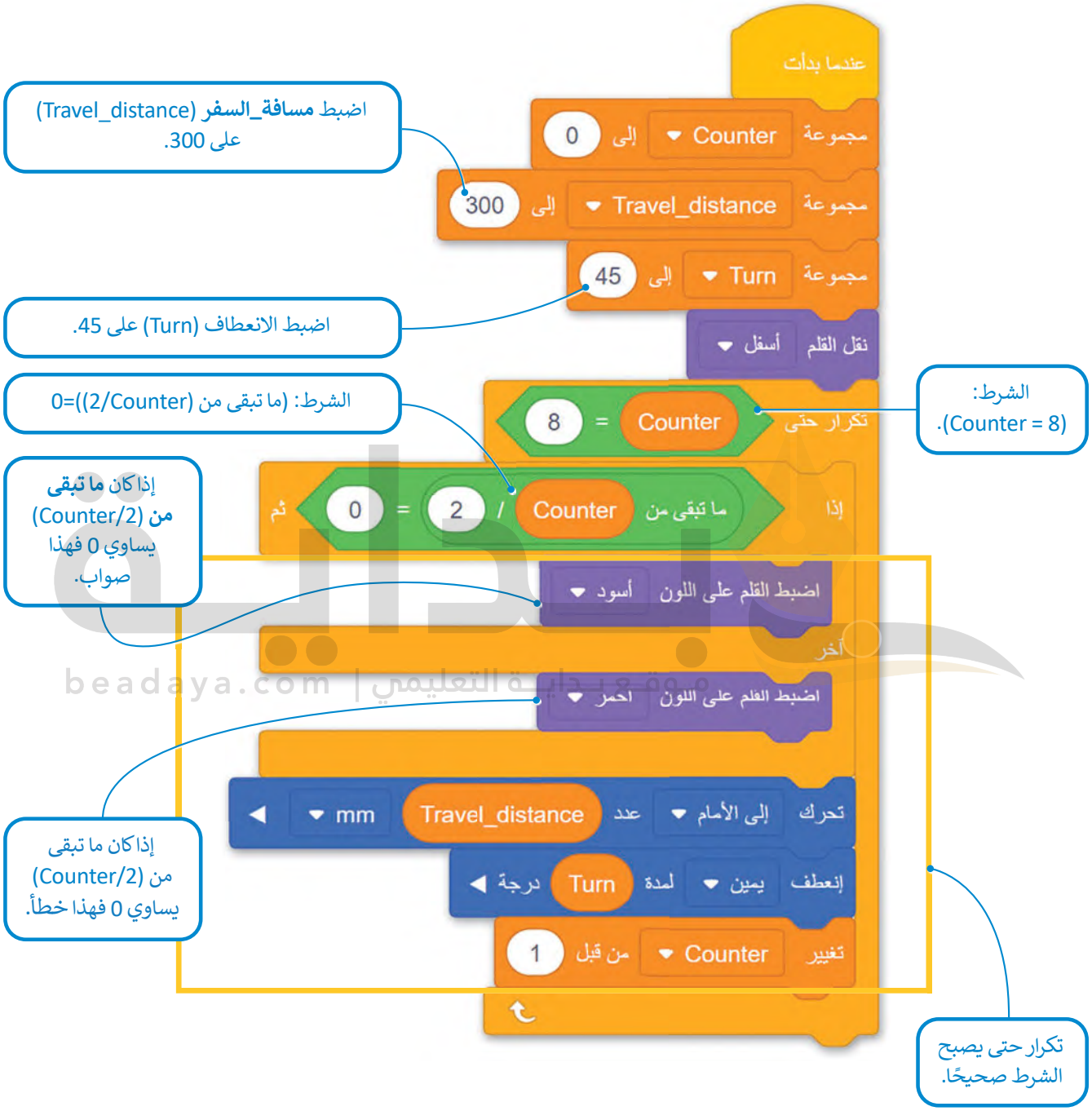
تستخدم لبنة **ما تبقى من ( ) / ( )** (remainder of ( ) / ( )) لقسمة القيمة الأولى على القيمة الثانية ثم عرض الباقي، ويمكنك العثور عليها في فئة **العمليات** (Operators).



عند قسمة عدد فردي على 2 سيكون الباقي دائمًا 1، بينما لن يكون للعدد الزوجي باقي عند قسمته على 2.

beadaya.com | موقع بداية التعليمي

برمج روبوت الواقع الافتراضي لرسم شكل ثماني.

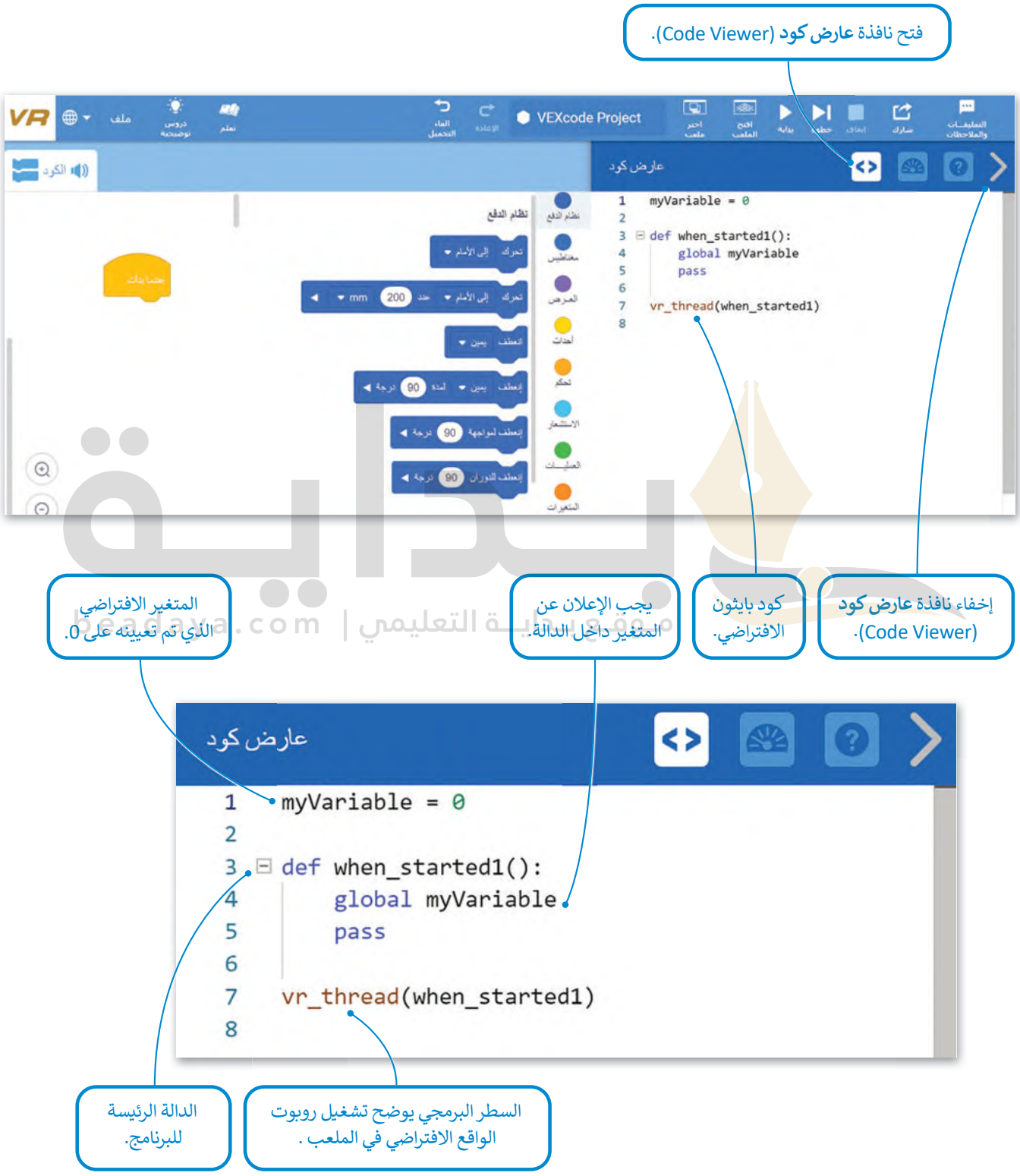


beadaya.com | موقع بداية التعليمي

عارض الكود (Code Viewer)

عند إنشاء مشروع يتكون من لبنات، يمكنك رؤية كود المشروع بلغة بايثون في نافذة **عارض كود** (Code Viewer).

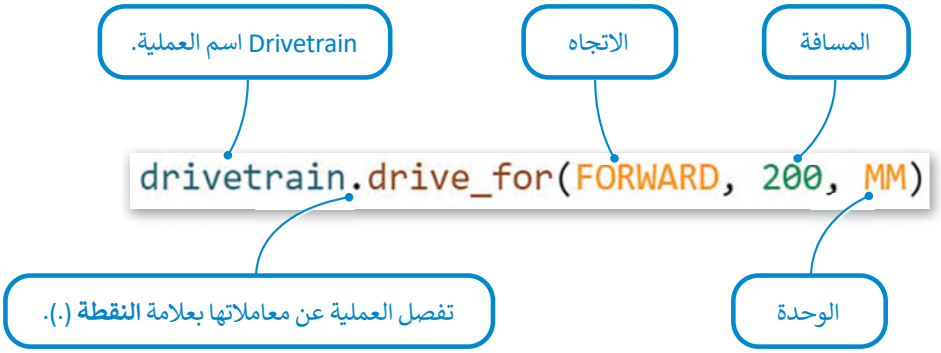
يسمح لك عارض الكود برؤية اللبنات والنصوص البرمجية في نفس الوقت، وبهذه الطريقة يساعدك على فهم طريقة ترجمة كل لبنة إلى كود نصي في بايثون.



بداية للتعليم | beadaya.com

معاملات بايثون (Python parameters)

عند استخدام اللبنات البرمجية في بيئة فيكس كود في آر، يمكنك تغيير معاملاتها عن طريق اختيار أحد الخيارات المختلفة من القائمة المنسدلة أو تغيير الأرقام داخل المساحة الدائرية، ولكن في بايثون تستخدم الفواصل للفصل بين المعاملات المختلفة.



يعرض الجدول التالي خمس لبنات أساسية وأوامر بلغة بايثون في بيئة فيكس كود في آر:

لبنات في بيئة فيكس كود في آر	أوامر بايثون في بيئة فيكس كود في آر
	`drivetrain.drive_for(FORWARD, 200, MM)`
	`drivetrain.turn_for(RIGHT, 90, DEGREES)`
	`drivetrain.set_drive_velocity(50, PERCENT)`
	`for repeat_count in range(10):`
	`wait(1, SECONDS)`

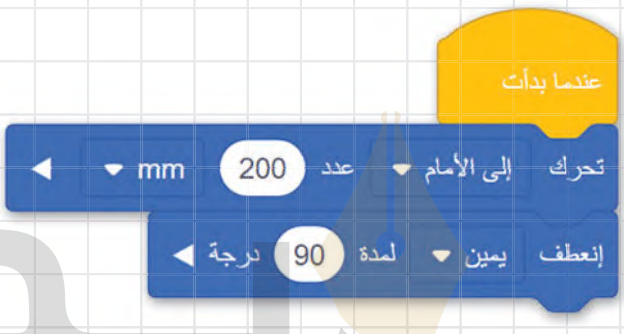
بداية
موقع بداية التعليمي

قبل تجربة المثال، احذف المتغير الافتراضي من فئة المتغيرات (Variables) لمسح الكود الخاص بك.

مثال 8: الحركة

في هذا المثال، يمكنك أن ترى كيف تمت كتابة لبنتين أساسيتين لحركة روبوت الواقع الافتراضي بلغة بايثون في نافذة عارض كود.

سيتحرك الروبوت للأمام لمسافة 200 مليمتر ثم الانعطاف 90 درجة يمينًا.




```
def when_started1():
    drivetrain.drive_for(FORWARD, 200, MM)
    drivetrain.turn_for(RIGHT, 90, DEGREES)

vr_thread(when_started1)
```

beadaya.com

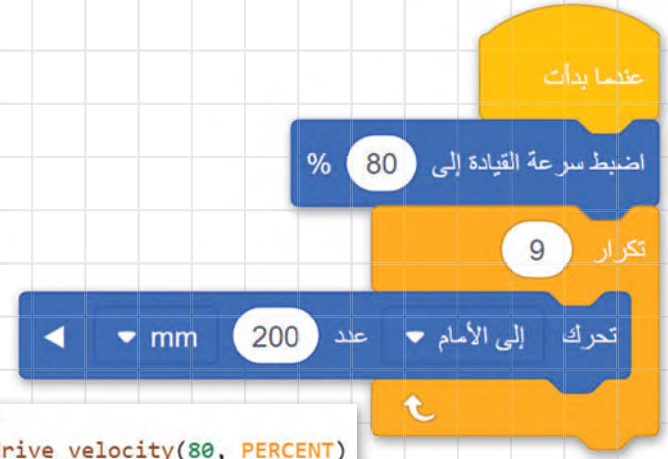
حلقة For

يتم استخدام حلقة for عندما تريد تكرار مجموعة من الأوامر لعدد محدد من المرات، ويتم تحديد عدد التكرارات في معامل **النطاق ( )** (range ( )).

مثال 9: تكرار الحركة

في هذا المثال، ستكرر الحلقة for الخطوات التي تم وضع مسافة بادئة لها 9 مرات.

سيتحرك الروبوت بسرعة 80 %، ويتقدم للأمام مسافة 200 مليمتر، ثم يكرر ذلك 9 مرات.




```
1  def when_started1():
2      drivetrain.set_drive_velocity(80, PERCENT)
3      for repeat_count in range(9):
4          drivetrain.drive_for(FORWARD, 200, MM)
5          wait(5, MSEC)
6
7  vr_thread(when_started1)
```

يجب أن تكون هناك مسافة بادئة قبل العبارات المكررة.

يتم إضافة تأخير 5 ملي ثانية بشكل افتراضي عند استخدام الحلقة.

حلقة While

يتم استخدام حلقة while عندما لا يكون عدد التكرارات معروفًا. عندما يكون الشرط صحيحًا فإن الحلقة تتكرر، ثم يتم فحص الشرط بعد كل تكرار. وعندما يكون الشرط خطأ يتوقف التكرار ويُنفذ السطر الذي يلي الحلقة في البرنامج. أما إذا كان الشرط خطأ من البداية فلن يتم تنفيذ عبارات الحلقة على الإطلاق.

مثال 10: العدّ

في هذا المثال، ستنشئ متغيرًا باسم Counter، ويتم تعيينه على 0 في بداية البرنامج، ثم يضيف البرنامج 1 حتى تكون قيمة المتغير Counter أكبر من 5.


```
1   Counter = 0
2
3   def when_started1():
4       global Counter
5       Counter = 0
6       while not Counter > 5:
7           Counter = Counter + 1
8           wait(1, SECONDS)
9
10  vr_thread(when_started1)
```

لنطبق معًا

تدريب 1

قواعد اسم المتغير في فيكس كود في آر.

حدد الجملة الصحيحة والجملة الخطأ فيما يلي:	صحيحة	خطأ
1. في هذا البرنامج يجب أن يكون اسم المتغير فريدًا.	✓	
2. كل كلمة يمكن أن تكون اسم متغير.		✓
3. قد يحتوي اسم المتغير على أحرف خاصة.		✓
4. قد يحتوي اسم المتغير على مسافات.		✓
5. قد يتكون اسم المتغير من مجموعة من الأحرف الكبيرة والصغيرة.	✓	

بداية
beadaya.com | موقع بداية التعليمي

تدريب 2

اكتب رقم اللبنة البرمجية أمام الأمر الصحيح بلغة بايثون.


```
speed = 20
```
2

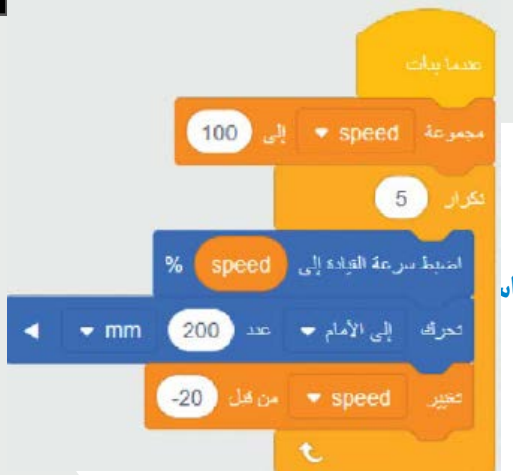
```
drivetrain.set drive velocity(20, PERCENT)
```
1

```
drivetrain.drive_for(FORWARD, 300, MM)
```
4

```
    for repeat_count in range(20):
```
3

```
drivetrain.drive_for(REVERSE, 300, MM)
```
5

بداية
beadaya.com | موقع بداية التعليمي

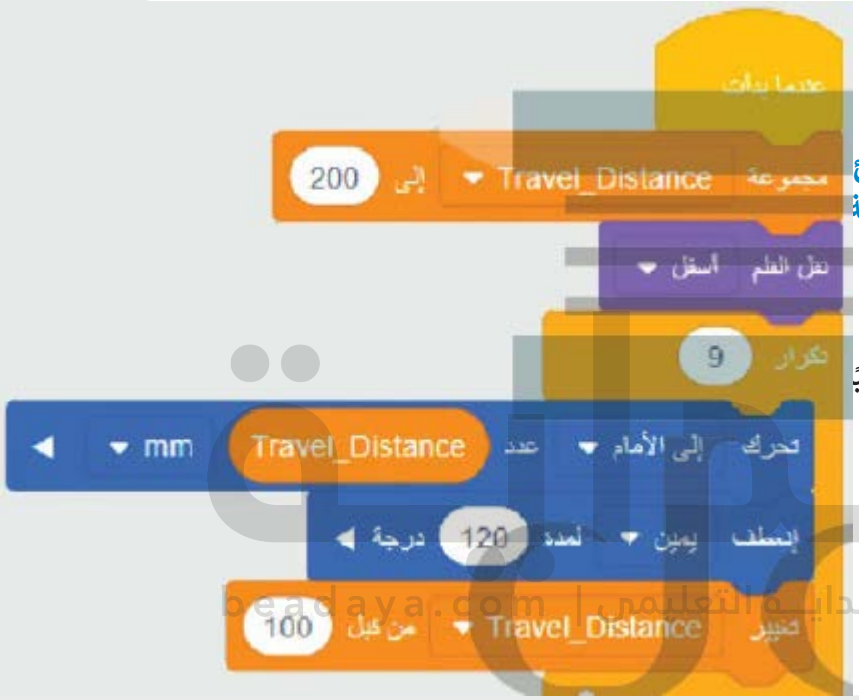


تدريب 3

بناءً على الكود الذي أنشأته في مثال التسارع، أجر التغييرات المنا... المرة.

> يجب أن تكون سرعة بدء الروبوت 100.

تدريب 4



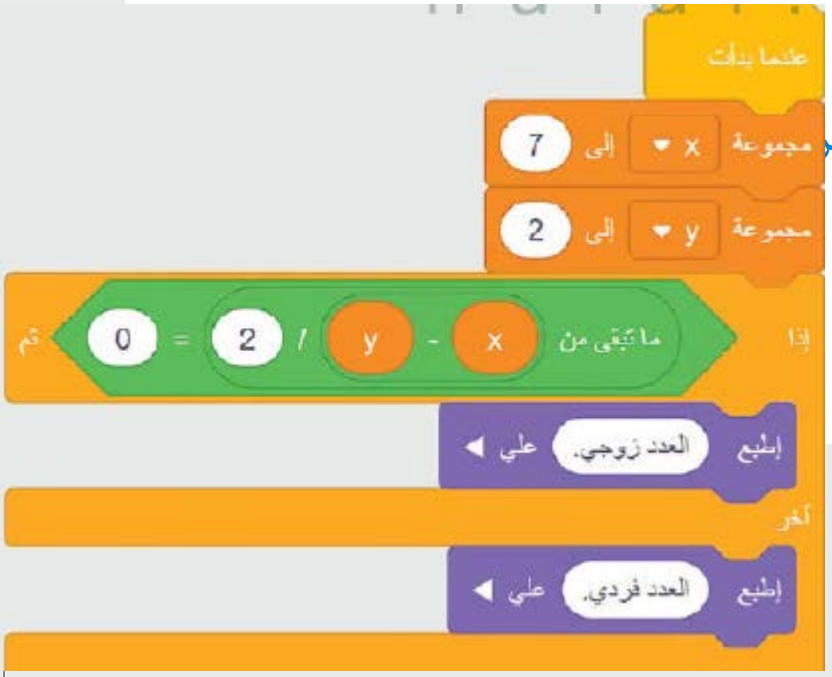
استخدم ساحة لعب الفن قماش، وأنشئ... يرسم فيه الروبوت ثلاثة مثلثات حلزونية... موضح في الصورة.

> استخدم متغيرًا للتكرار.

> تذكر أنه في كل مرة يرسم فيها الروبوت جانبً... يجب أن يكون أكبر من الجانب الذي قبله.

بدايا
موقع بدايا التعليمي | beadaya.com

تدريب 5



أنشئ برنامجًا لتحديد ما إذا كانت نتيجة ط... الرسائل التالية:

> العدد فردي.

> العدد زوجي.